

00 Código	Facultad	Carrera	Asignatura
	T	T4	09

01 Facultad: Tecnología Informática **Carrera:** Analista Programador

02. Asignatura: Programación orientada a objetos

03 Año lectivo: 2021

04 Año de Cursada: 2º

05 Cuatrimestre: 1º

06 Hs. Semanales: 6

07 Profesor Titular: Darío G. Cardacci

Director de Área: Darío G. Cardacci

Tutores: Darío G. Cardacci

08 Ítems del perfil que se desarrollarán:

Los sistemas utilizados en la actualidad demandan para su administración la utilización de paradigmas que se adapten a una realidad cambiante, así como acceder a grandes volúmenes de datos. Es por ello que el profesional que realice esta tarea debe estar entrenado en implementar tecnologías de punta orientadas al desarrollo de sistemas de información.

Para ello, los paradigmas utilizados en el desarrollo de los nuevos sistemas de información, deben poseer las características necesarias que permitan representar virtualmente en un ámbito computacional los problemas reales que requieren una solución.

Otro aspecto a incentivar es la capacidad para poder actuar con eficiencia y eficacia ante situaciones que requieran pericia en la materia.

Un aspecto importante a tener en cuenta es el aporte de habilidades para lograr el trabajo en grupo, la integración, la capacidad de transferir conocimientos y generar una visión crítica sobre la realidad que afecta al desarrollo de software.

09 Correlativas **previas:** Programación I
 posteriores: Lenguajes de Última Generación

10 Articulación con asignaturas del mismo año:

No están previstas actividades de articulación con otras asignaturas

11 Objetivos:

- Incorporar los fundamentos de la programación orientada a objetos.
- Ampliar y profundizar los conceptos teóricos del modelo de orientación a objetos.
- Conocer las instrucciones que permiten gestionar la depuración de programas.
- Construir programas de aplicación visual que acceden y procesan datos.
- Construir programas orientados a objetos que acceden y establecen comunicaciones entre aplicaciones.
- Crear componentes de software cliente-servidor.
- Incorporar tecnologías aplicadas en Internet a los desarrollos de software aplicativo.
- Diseñar programas para manejar equipamiento a distancia.
- Construir programas orientado a objetos con interfaz de aplicación visual.

12 Unidades de desarrollo de los contenidos:

Unidad 1: OBJETOS Y CLASES.

El modelo orientado a objetos.

Jerarquías “Es - Un” y “Todo - Parte”.

Concepto de Objeto.

Características básicas de un objeto: estado, comportamiento e identidad.

Ciclo de vida de un objeto.

Modelos. Modelo estático. Modelo dinámico. Modelo lógico. Modelo físico.

Concepto de análisis diseño y programación orientada a objetos.

Conceptos de encapsulado, abstracción, modularidad y jerarquía.

Concurrencia y persistencia.

Concepto de clase.

Definición e implementación de una clase

Campos y Constantes

Propiedades. Concepto de Getter() y Setter(). Propiedades de solo lectura. Propiedades de solo escritura. Propiedades de lectura-escritura. Propiedades con indizadores. Propiedades autoimplementadas. Propiedades de acceso diferenciado.

Métodos. Métodos sin parámetros. Métodos con parámetros por valor. Métodos con parámetros por referencia. Valores de retorno de referencia.

Sobrecarga de métodos.

Constructores. Constructores predeterminados. Constructores con argumentos.
Finalizadores.
Clases anidadas.

Tiempo: 24 Hs.

Unidad 2: OBJETOS Y CLASES - RELACIONES

Eventos.
Suscripción a eventos. Suscripción a eventos utilizando el IDE. Suscripción a eventos mediante programación. Suscripción a eventos mediante métodos anónimos.
Publicación de eventos.
Desencadenar eventos.
Modificadores de acceso.
Clases abstractas, selladas y estáticas. Miembros estáticos en clases estáticas.
Sobrescritura de métodos. Métodos virtuales.
Relaciones básicas entre clases. “Generalización-Especialización”, “Parte de”.
Relaciones derivadas entre clases. Herencia. Herencia simple. Herencia múltiple. Polimorfismo.
Agregación. Simple y con contención física.
Asociación y relación de Uso.
Teoría de Tipos. Tipos anónimos.
Elementos que determinan la calidad de una clase: acoplamiento, cohesión, suficiencia, compleción y primitivas.
Relaciones entre objetos: enlace y agregación.
Acceso a la clase base desde la clase derivada.
Acceso a la instancia actual de la clase.

Tiempo: 24 Hs.

Unidad 3: FRAMEWORKS Y MANEJO DE EXCEPCIONES.

Concepto de frameworks. Elementos de un framework. Tipos de frameworks.
Arquitectura de .NET. Interoperatividad entre .NET y COM.
Código administrado y no administrado.
Common Language Runtime CLR.
Lenguaje intermedio IL.
El compilador Just-in-Time (JIT).
Concepto de assembly.
Administración de la memoria en .NET. El Garbage Collector.
Manejo de excepciones. Control de excepciones. El objeto Exception. La instrucción Try – Catch – Finally. La instrucción Throw.
Depuración de aplicaciones. Herramientas de depuración. Análisis del comportamiento de las aplicaciones.

Tiempo: 12 horas

Unidad 4: INTERFACES Y DELEGADOS

Interfaces. Desarrollo e Implementación de una interfaz.

La interfaz IComparable. La interfaz IComparer. La interfaz ICloneable. Las interfaces IEnumerable e IEnumerator.

Delegados. Delegados con métodos con nombre. Delegados con métodos anónimos.

Tiempo: 12 horas

Unidad 5: GENERICOS, LINQ Y EXPRESIONES LAMBDA

Introducción a los genéricos. Ventaja de usar genéricos. Clases genéricas. Interfaces y métodos genéricos.

LINQ to Object. Introducción a las consultas con LINQ. Escritura de consultas con LINQ. Retorno y almacenamiento de consultas LINQ. Grupos anidados y subconsultas con LINQ.

Introducción a las expresiones lambda. Funciones anónimas. Uso de expresiones lambda en consultas.

Tiempo: 12 horas

Unidad 6: COMUNICACIÓN ENTRE APLICACIONES Y MANEJO DE DISPOSITIVOS

Esquema cliente - servidor. Configuración de aplicaciones remotas en una red. Formas de compartir información entre aplicaciones. Pasaje de información batch vs. On line.

Conceptos básicos de protocolos. TCP. UDP. Concepto sobre IP, TCP/IP. Concepto de software cliente – servidor. Ventajas y desventajas. Distribución de procesos y Almacenamientos. Sistema de mensajería de un sistema cliente – servidor.

Concepto de Subproceso. Utilización de Sockets. Sockets de clientes sincrónicos y asincrónicos. Socket de servidores sincrónicos y asincrónicos.

Tiempo: 12 hs.

13 Metodología de trabajo:

Durante toda la cursada promoveremos la articulación e integración de los contenidos a través de actividades que remitan a la puesta en práctica de los conceptos teóricos.

Se tendrá especial atención a la participación de los alumnos en los espacios de intercambio y colaboración, para poder rescatar la elaboración conceptual que van construyendo acerca de los temas tratados.

Los medios audiovisuales ocuparán un importante lugar en la enseñanza de los contenidos para favorecer otros modos de representación del conocimiento

Los textos de actualidad y las notas periodísticas sobre tecnología de punta serán propuestos para ser leídos y tratados en la cursada.

Se fomentará la participación de los alumnos en la formulación de soluciones a los problemas de la práctica profesional.

Se desarrollarán ejercicios que denoten un alto grado de relación grupal para aprovechar la sinergia de la relación interpersonal.

Se alentará el desarrollo de investigaciones sobre temas abordados en la asignatura.

Herramientas: Lenguaje de programación Visual Orientado a Objetos.

14 Trabajos prácticos:

La asignatura contempla la realización de dos trabajos prácticos.

Estos trabajos prácticos junto con otros indicados por el docente formarán parte de una carpeta individual de ejercicios prácticos que será entregada en la fecha de la evaluación final. La forma de entrega es impresa y en soporte magnético.

15 Bibliografía:

Obligatoria:

Cardacci Darío y Booch, Grady. 2013. **Orientación a Objetos. Teoría y Práctica**. Buenos Aires, Argentina. Pearson Argentina, 2013.

Deitel Harvey M. Y Paul J. Deitel. 2007. **Cómo programar en C#**. Segunda edición. México.

Lenguaje C#. [https://msdn.microsoft.com/es-es/library/aa287558\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa287558(v=vs.71).aspx)

Tutoriales C#. [https://msdn.microsoft.com/es-es/library/aa288436\(v=vs.71\).aspx](https://msdn.microsoft.com/es-es/library/aa288436(v=vs.71).aspx)

Ampliatoria:

Martin, James; Odell, James J. 1997. **Métodos orientados a objetos: conceptos fundamentales**. México, DF. Prentice Hall Hispanoamericana.

Montefinal, Fabián H.; Cardacci, Darío G. 2006. **Conceptos básicos sobre electricidad: electrónica y puertos de la PC**. 2a.ed. Buenos Aires: Universidad Abierta Interamericana.

Rumbaugh, James; Blaha, Michael; Premerlani, William y otros. 1991. **Modelado y diseño orientado a objetos**. Madrid. Prentice Hall Regent.

Zacker, Craig Rourke, John. 2001. **PC Hardware: manual de referencia**. Madrid. McGraw-Hill Interamericana.

16 procedimiento de evaluación y criterio de promoción:

Parciales

Los exámenes parciales serán como mínimo dos, cada uno con teoría y práctica. Ambos son obligatorios, individuales y escritos.

Los parciales deberán tener un 60% de contenido correcto para estar aprobados. En este último caso la nota deberá quedar establecida entre 4 (cuatro) y 10 (diez) puntos. Si el alumno no realiza un examen parcial se calificará con una nota insuficiente.

Recuperatorios

Los exámenes recuperatorios se desarrollarán a razón de uno por cada parcial con teoría y práctica. Se podrá optar por tomar un recuperatorio integral en aquellos casos que se justifique y el mismo deberá ser equivalente al alcance de los recuperatorios individuales.

Recuperarán los parciales que correspondan, aquellos alumnos que obtuvieron una calificación inferior a 4 (cuatro) puntos en alguno de sus exámenes parciales. Los recuperatorios son individuales y por escrito. El docente dispondrá la fecha para realizar esta actividad que podrá ser con posterioridad a cada uno de los parciales o en una fecha común para todos. Las notas de los exámenes recuperatorios no reemplazan las calificaciones insuficientes obtenidas en los exámenes regulares que se están recuperando, las mismas serán tomadas como parte de las calificaciones que intervienen en el promedio.

Trabajos prácticos.

Se desarrollarán los previstos en el punto 14.

Carpeta de ejercicios.

La carpeta de ejercicios será un insumo para la nota del final.

Nota conceptual.

Una nota conceptual basada en la participación y colaboración del estudiante, el interés demostrado en la asignatura y el empeño puesto de manifiesto en las actividades de estudio.

Otros aspectos generales de la evaluación.

Los cuestionarios de autoevaluación elaborados para el curso poseen como objetivo el desarrollo de un espíritu crítico en el alumno sobre su desempeño en la futura práctica profesional.

Criterios para la promoción.

Para aprobar la asignatura y acceder al examen final o coloquial el alumno deberá tener presentados y corregidos sus dos exámenes parciales y los dos Trabajos Prácticos Requeridos.

El alumno deberá tener como mínimo cuatro notas para poder promediar sin contar los recuperatorios de parciales.

Las notas para aprobar la asignatura y acceder al examen final o coloquial serán las que rijan para la cátedra de acuerdo a lo que dictamine la facultad